Introduction

This module explores several strategies that can be used for searching and sorting a list of elements. A strategy is an algorithm or a set of instructions guiding the interpreter to transform given inputs into useful information that is required by the user to solve a particular problem. After covering how to design these algorithms in Python, the module will show the search and sort algorithms' implementation by abstracting them using Python functions. A detailed analysis of the efficiency will be given with each algorithm, and an illustrative example will be provided. The Big-O notation will be used to measure the efficiency of the provided algorithms in terms of the number of steps rather than the units of time. The Big-O notation is a standard technique that helps categorize the algorithm's efficiency to achieve consistency when comparing one algorithm with another.

Topics and Learning Objectives

Topics

- Linear search
- Binary search
- Selection sort
- Bubble sort
- Insertion sort

Learning Objectives

By the end of the module, you will be able to:

- Construct a simple linear search algorithm.
- Distinguish between different sort algorithms.
- Quantify the efficiency of an algorithm.

Readings

Required Readings

- Lambert, K. A. (2019). *Fundamentals of Python: First programs* (2nd ed.). Boston, MA: Cengage Learning.
    - Searching, Sorting, and Complexity Analysis, pages 390-431

Search Algorithms

Please watch this video presentation as it highlights two search algorithms: **linear search** and **binary search**. Learning these two algorithms will help find a particular item inside a given data structure, such as a list. Each algorithm's design will be explored with an example to show how that algorithm works during runtime. Understanding each algorithm's structure helps to differentiate it and identify a suitable algorithm based on the problem's requirements and the algorithm's efficiency. For example, if we have a 100-element sorted list and the problem is finding the smallest value, then what is the most efficient strategy to find an answer and how can that strategy be employed in Python?

Rycast Video:
CIND830 - Module 9: Search Algorithms

Download

Download the PDF file for **CIND830 - Module 9: Search and Sort Algorithms.**

Please note this PDF file will be used in the two videos of this module. In the PDF, you can use the top navigation bar to navigate between the topics covered in each video.

Download the **CIND830 - Module 9: Search Algorithms transcript**.

Practical Examples of Search Algorithms Using the Bisect Library

This video covers a binary search module named “bisect.” First, the main functions of the library are explained. Then, two examples are illustrated to demonstrate how this module can be used to solve search problems.

Rycast Video:
CIND830 - Module 9: Practical Examples of Search Algorithms Using the Bisect Library

Download

Download the **CIND830 - Module 9: Practical Examples of Search Algorithms Using the Bisect Library transcript**.

Sort Algorithms

Please watch this video presentation as it demonstrates three algorithms for sorting a particular data structure, such as a list. These algorithms are: **selection sort, bubble sort and insertion sort**. Each one of these algorithms operates on either an ordered or unordered list of integers. First, the below video will explain the algorithm's design, then give a practical example to explore the algorithm's mechanism. The computational complexity will be explained, as it is useful in contrasting algorithms and determines which one is efficient enough to be applied for a given scenario.

Rycast Video:
CIND830 - Module 9: Sort Algorithms

Insert Download box

Download the PDF file for **CIND830 - Module 9: Search and Sort Algorithms** and click on the topic “**Sort Algorithms**” in the top navigation bar to take you to the beginning of this topic.

Download the **CIND830 - Module 9: Sort Algorithms transcript**.

Practical Examples of Sort Algorithms Using the Bisect Library

This video explains how to sort different data types in Python language. Firstly, we introduce standard sort functions for the most common data types in Python language such as lists and tuples. Secondly, sorting custom objects is explained. Finally, we discuss the time complexity of the sort and sorted functions in Python.

Ryecast Video:
CIND830 - Module 9: Practical Examples of Sort Algorithms Using the Bisect Library

Insert Download box

Download the **CIND830 - Module 9: Practical Examples of Sort Algorithms Using the Bisect Library transcript**.

Self-Assessment Quiz

Please complete the following review activity. The activity is not graded.

Test Yourself

H5P Object:
Self-Assessment Quiz

Summary

In this module, we explored five strategies for searching and sorting a list of integers. We showed the abstraction and implementation along with detailed performance analyses in terms of the standard Big-O notation. The Big-O notation is a useful measure, as it helps in contrasting algorithms. For example, if two algorithms come with different orders, we can decide which one to use with a large or small number of elements, as one of those algorithms should be faster than the other, at least at a certain point or with a particular number of elements.

Activities

Discussion

Module 9 Discussion

This week, let's discuss search and sort algorithms in the **Module 9 Discussion** topic in D2L under the **Communication** drop-down menu, then **Discussions**. Feel free to start a new thread or respond to someone else's post.

Activity

Lab 7

Before starting Lab 7, be sure you have reviewed all the content in Module 9. The lab is a practical extension to apply your knowledge gained from this module. Download Lab 7 now and complete it by the end of this week. Your instructor will give you time to work on it before releasing the solutions in a folder called **Lab Solutions** in D2L under the **Course Materials** drop-down menu for you to verify your answers. The lab activity is not graded.

While working on your lab, post any questions, ideas, and strategies in the **Module 9 Discussion** topic in D2L. Feel free to read and respond to your peers' posts as well. You may also have the option of reaching out to your instructor during his/her virtual office hours.

Please note: The PDF for the lab activity may not be fully accessible for students who use assistive technology. If you require an accessible version, please contact your instructor.

Assessments

Quiz Module 9

Please go to your D2L course shell, under the **Assessment** drop-down menu, select **Quizzes**. Then click on the **Quiz Module 9 link** and follow the instructions to complete this quiz. The date and length of time have been set for this quiz. **This quiz is a cumulative 10% of your final grade.**

Assignment 2

Assignment 2 is worth 15% of your final grade and due by the end of Module 12. Remember to submit both the notebook file (in IPYNB format) and the exported version (either in PDF or HTML format) in the same **Assignment 2 link** in D2L. Failing to submit both files will be subject to a mark deduction.

While you are working on this assignment, feel free to share your questions, ideas and strategies in the **Assignment 2 Discussion** topic in D2L.

References

- Lambert, K. A. (2019). *Fundamentals of Python: First programs* (2nd ed.). Boston, MA: Cengage Learning.
- Python Software Foundation. (2020). Python 3.7.8 documentation. Retrieved from https://docs.python.org/3.7/index.html.